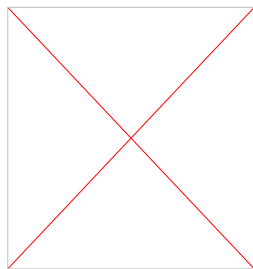


Reducing the zero results rate for search on Wikipedia

28th August 2015



Dan Garry
Lead Product Manager
@danjgarry



Discovery Department

Building the anonymous path of discovery to a trusted and relevant source of knowledge.



Yuri
Astrakhan



Erik
Bernhardson



David
Causse



Tomasz
Finc



Dan
Garry



Trey
Jones



Oliver
Keyes



Stas
Malyshev



Wes
Moran



Mikhail
Popov



Max
Semenik



Kevin
Smith



Moiz
Syed



Previously, on
“Presentations by Dan”...

This quarter, we're dedicated to
reducing the zero results rate.

Users should never get zero results!

And now...
the continuation...

Who's getting zero results
anyway? And why?

We analysed the queries that
got zero results to identify
usage patterns.

Here's some examples.

Follow along: bit.ly/zeroreresults

#1: DOI queries

"10.3897/zookeys.457.6760" OR "<http://zookeys.pensoft.net/articles.php?id=4267>"

"10.3332/ecancer.2013.301"

"10.7821/naer.1.1.2-6"

#1: DOI queries

An app called Lagotto that checks
if Wikipedia has a citation for a paper.

Giving them zero results a lot is legitimate, as
most scientific papers are not cited.

Apparently the volume of requests
may be a configuration issue!


#2: Unix timestamps

1431786835781: بيت لحم


1436436482196: Илюзия

1432198699732: Meryl Streep



Change-Id: I64b6ae2d72caa9c535f6a2a5627dc7a23eb2f6f7 

Owner Deskana

Project  [apps/android/wikipedia](#)


Branch master

Topic T96944

Uploaded Apr 29, 2015 9:45 PM

Updated Apr 30, 2015 9:41 AM

Status Merged

★ **Commit Message** [Permalink](#) 

Make some PersistenceHelpers set the namespace of pages as appropriate

Currently, SavedPagePersistenceHelper and HistoryEntryPersistenceHelper do not set the namespace of a page appropriate, setting it to null. I'm not sure why it was implemented that way.

This patch makes SavedPagePersistenceHelper and HistoryEntryPersistenceHelper set the namespaces of pages appropriately.

Fortunately, the method in which the database is upgraded ensures the old entries are still compatible with the new system. They'll continue to suffer the same problem where the pages are constructed with null namespaces, but otherwise they'll work as intended without any tweaking.

Bug: [T96944](#)

Change-Id: [I64b6ae2d72caa9c535f6a2a5627dc7a23eb2f6f7](#)

Reviewer	Code-Review	Verified
Deskana		
Dbrant		
Brion VIBBER		
BearND	✓	
jenkins-bot		✓

“...the old entries are still compatible with the new system.”

Famous last words.

#3: title AND link_from_article

"Argentine_football_league_system" AND "Football in Moldova"

"Argentine_football_league_system" AND "Football in Mongolia"

"Argentine_football_league_system" AND "Football in Mozambique"

"Argentine_football_league_system" AND "Football in Papua New Guinea"

#3: title AND link_from_article

All these come from a single IP, and it accounted for around 5% of total search traffic.

The sheer volume of requests totally skewed one of our A/B tests.

We had to block him. :-)

There's still a lot of zero queries
not falling into these buckets.

How do we help them?

Patterns for attacking the zero results problem


1. Query forwarding
 - Change the user's query to one more likely to find what they want
 - Run the user's query against a different data source
2. Elasticsearch parameter improvements
 - Optimise parameters to return better results for users
3. Use different backends for search results
 - Generate search results using totally different algorithms

Building the infrastructure

1. Write lightweight A/B testing system for CirrusSearch
 - Bucket users according to defined sampling ratio
 - Give bucketed users different search parameters
2. Create a test environment to try crazy new things
 - Import entire enwiki search index into a testwiki in Labs
 - Replay queries against the test index and measure the results
3. Dashboard and analysis
 - Make a search dashboard: <https://searchdata.wmflabs.org>
 - Analyse A/B test results and produce reports of test outcomes

What have we done?

If the searcher gets zero results, and also a suggestion, just run the suggestion.





WIKIPEDIA
The Free Encyclopedia

[Main page](#)
[Contents](#)
[Featured content](#)
[Current events](#)
[Random article](#)
[Donate to Wikipedia](#)
[Wikipedia store](#)

Interaction
[Help](#)
[About Wikipedia](#)
[Community portal](#)
[Recent changes](#)
[Contact page](#)

Tools
[Upload file](#)
[Special pages](#)
[Printable version](#)

Languages 

[Special page](#) 

Search results

[Help:Searching](#)

[Search](#) Results 1 - 20 of 74,279

Content pages [Multimedia](#) [Everything](#) [Advanced](#)

Showing results for **manchester**. Search instead for **manchesterr**.

*The page "**Manchesterr**" does not exist. You can [ask for it to be created](#), but consider checking the search results below to see whether the topic is already covered.*

Manchester
Manchester in England. For the larger conurbation, see Greater **Manchester** Built-up Area. For the wider metropolitan county, see Greater **Manchester**.
140 KB (13,299 words) - 05:48, 27 August 2015

Manchester United F.C.
see **Manchester** United (disambiguation). **Manchester** United Football Club is a professional football club based in Old Trafford, Greater **Manchester**, England
107 KB (10,105 words) - 13:21, 27 August 2015

Greater Manchester
Greater **Manchester** is a metropolitan county in North West England, with a population of 2.7 million. It encompasses one of the largest metropolitan areas
165 KB (16,003 words) - 20:40, 24 August 2015

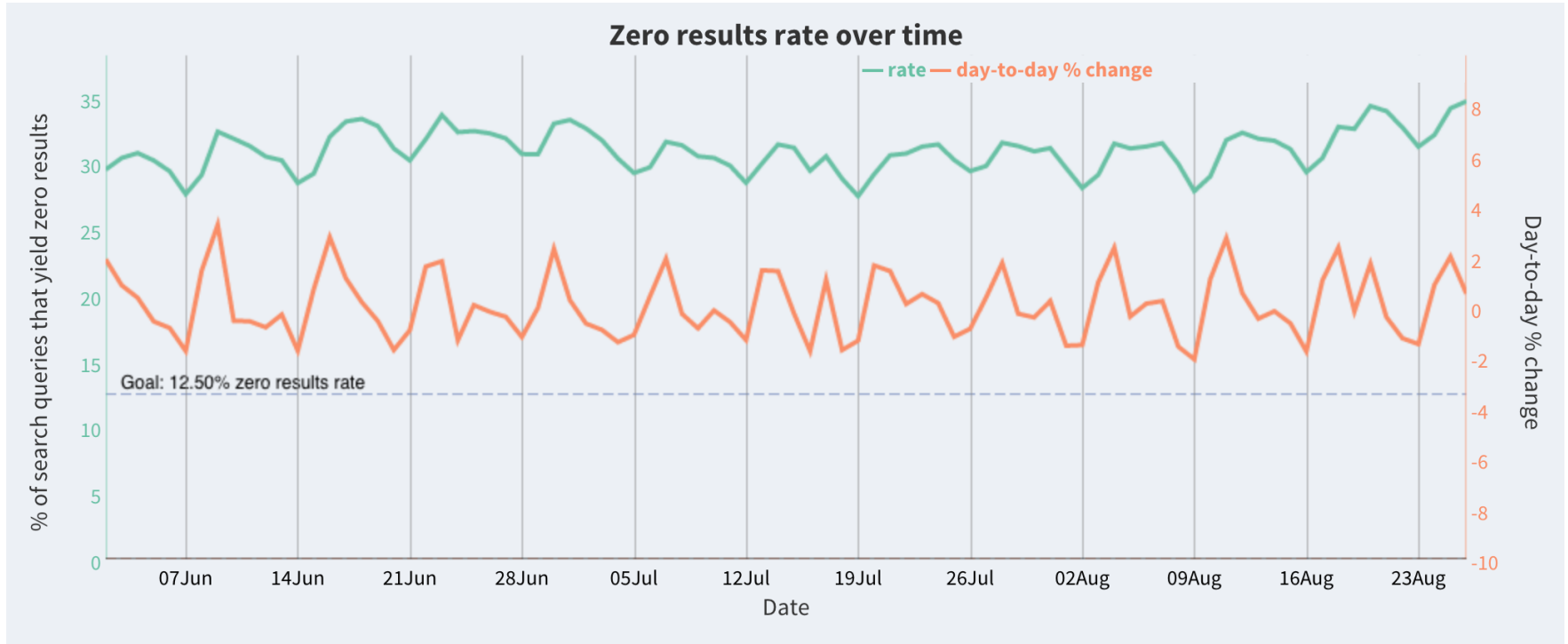
What have we done?

Run A/B tests to figure out if there are better search parameters to use.

<http://bit.ly/zeroABtest>

```
15252 'wmgCirrusSearchUserTesting' => array(  
15253   'default' => array(  
15254     'suggest-confidence' => array(  
15255       'sampleRate' => 10,  
15256       'buckets' => array(  
15257         // control bucket, retain defaults  
15258         'a' => array(),  
15259         // test bucket, alternative suggestions  
15260         'b' => array(  
15261           'wgCirrusSearchPhraseSuggestSettings' => array(  
15262             'mode' => 'always',  
15263             'confidence' => 1.0,  
15264             'max_errors' => 2,  
15265             'real_word_error_likelihood' => 0.95,  
15266             'max_term_freq' => 0.5,  
15267             'min_doc_freq' => 0.0,  
15268             'collate' => false,  
15269             'collate_minimum_should_match' => '3<66%',  
15270             'smoothing_model' => array(  
15271               'laplace' => array(  
15272                 'alpha' => 0.3,  
15273               ),  
15274             ),  
15275           ),  
15276         ),  
15277       ),  
15278     ),  
15279   ),  
15280 )
```

How is the zero results rate looking?



What's next?

We need to try something more radical
to achieve our goals.

**Why don't we generate our search results
a completely different way?**

Elasticsearch Completion Suggester



Completion suggester experiment

Search: Completion: 1148ms

- Jurassic World
- Jurassic Park
- Jurassic
- Jurassic Park III
- Jurassic Park II
- Jurassic Exxplosion Phillipic

Prefix: 1228ms

- Jurassic** park
- Jurassic**
- Jurassic** world

Elasticsearch Completion Suggester

Is the completion suggester
a viable alternative to prefixsearch?

Replaying unsampled zero results enwiki queries

API	results	no results	zero %
prefix_search	0	100000	100%
completion_suggest	38295	61731	62%

Replaying sampled enwiki queries

API	results	no results	zero %
prefix_search	288210	134469	32%
completion_suggest	340562	82183	19%

Elasticsearch Completion Suggester

Initial tests are promising, showing the completion suggester cutting zero results rate by nearly 40%.

But how can we *really* know?

Elasticsearch Completion Suggester

We've deployed the completion suggester API to production.

This doesn't change search, it just lets us run tests on the suggester.

Now we need to A/B test the suggester to see if it's better.

Thank you!

- Scrutinise our dashboards: <http://searchdata.wmflabs.org/>
- Ask us questions: wikimedia-search@lists.wikimedia.org